The Practicalities of Goal-Based Safety Regulation

J Penny, A Eaton CAA (SRG)

PG Bishop, RE Bloomfield (Adelard)

1 Introduction

"Goal-based regulation" does not specify the means of achieving compliance but sets goals that allow alternative ways of achieving compliance, e.g. "People shall be prevented from falling over the edge of the cliff". In "prescriptive regulation" the specific means of achieving compliance is mandated, e.g. "You shall install a 1 meter high rail at the edge of the cliff".

There is an increasing tendency to adopt a goal-based approach to safety regulation, and there are good technical and commercial reasons for believing this approach is preferable to more prescriptive regulation. It is however important to address the practical problems associated with goal-based regulation in order for it to be applied effectively.

This paper will first discuss the motivation for adopting a goal-based regulatory approach, and then illustrate the implementation by describing SW01 which forms part of the CAP 670 [CAA 1998] regulations for ground-based air traffic services (ATS). We will then discuss potential barriers to the implementation of such standards and how these difficulties can be addressed.

2 Regulatory context and motivation

The Robens Report [Robens 1972] and the Cullen Enquiry [Cullen 1990] were major drivers behind the UK Regulatory agencies exploring the benefits of introducing goal-based regulations. The reports noted several shortcomings with prescriptive safety regulations.

Firstly, with prescriptive regulations, the service provider is only required to carry out the mandated actions to discharge his legal responsibilities. If these actions then prove to be insufficient to prevent a subsequent accident, it is the regulations and those that set them that are seen to be deficient. Thus safety is viewed as the responsibility of the regulator and not the service provider whose responsibility, in law, it actually is.

Secondly, prescriptive regulations tend to be a distillation of past experience and, as such, can prove at best to be inappropriate and at worst to create unnecessary dangers in industries that are technically innovative. It is the innovator that is best placed to ensure the safety of their design, not the regulator. Clearly prescriptive safety regulations are unable to cope with a diversity of design solutions.

Thirdly, prescriptive regulations encode the best engineering practice at the time that they were written and rapidly become deficient where best practice is changing e.g. with evolving technologies. In fact it is quite probable that prescriptive regulations eventually prevent the service provider from adopting current best practice.

Another driver for adopting goal-based regulation, from a legal viewpoint, is that overly-restrictive regulation may be viewed as a barrier to open markets. Various international agreements, EC Directives and Regulations are intended to promote open markets and equivalent safety across nations. Whilst it is necessary to prescribe interoperability requirements and minimum levels of safety, prescription in other areas would defeat the aim of facilitating open markets and competition.

Finally, from a commercial viewpoint, prescriptive regulations could affect the cost and technical quality of available solutions provided by commercial suppliers.

So there are clear benefits in adopting a goal-based approach as it gives greater freedom in developing technical solutions and accommodating different standards. However, in order to adopt a goal-based approach, it is necessary to provide a coherent and convincing safety justification.

A common argument made to substantiate a claim that current best practice has been followed is to claim adherence to an international standard. This presents the regulator with a challenge as many safety assurance standards exist, coming from various industry sectors. Each one reflects the needs of its sector with respect to legal framework, supply chain characteristics and problem domain. Having encouraged open markets between sectors and countries, and given the service provider freedom of solution, he is fully entitled to call upon any of them. It is therefore necessary for the regulator to ensure that a parity of assurance is provided regardless of the assurance standard used. The regulator can address this issue by defining the credible bounds of an acceptable safety argument and the supporting evidence for each goal. The service provider is therefore required to argue how the evidence collected by using the assurance standard stays within these bounds and meets the goals. This offers an additional benefit of deterring service providers from preparing untenable arguments.

This then raises the question of what is sufficient argument and evidence to support a claim. Obviously the regulator gains confidence (that the claim is true) as the depth and strength (rigour) of the argument and evidence presented increases. The degree of confidence, that a regulator will require, varies with the risk attached to the claim being made. Such risks include the consequences of the claim being proved wrong and the believability of the claim. These risks are addressed by introducing the concept of Assurance Evidence Levels (AELs) to describe different levels of rigour of argument and evidence to be presented. It is important that the AEL is not confused with Safety Integrity Level (SIL), Design Integrity Level (DAL) or any other index used in assurance standards to identify appropriate practices for a particular integrity claim. Whilst such indexes are also risk based, they are used to help the service provider demonstrate compliance with best

practice, rather than to define what the regulator requires in order to have confidence that he can approve the system for use.

A goal-based approach can be applied at any level from the top-level system downwards. It is important that there are clear links between the top-level goals and the sub-goals. At each level, the regulator requires explicit safety *goals*, convincing *arguments* to justify the goals are met and adequate *evidence* to support the arguments. In practice the rigour of the arguments and the amount of evidence will depend on the safety significance of the individual system functions.

To illustrate the issues we will describe the approach adopted in the CAA Safety Regulation Group (SRG) for ground-based ATS systems that provide communications, surveillance and navigation services for aircraft. SRG regulate the safety of these services in three main ways:

- Approval of a safety management system. For large organisations like National Air Traffic Services (NATS), production and approval of safety cases is implemented "in-house" in accordance with the approved safety management procedures. The role of the SRG is to audit the implementation of safety management system and the associated safety cases.
- Approval of safety cases submitted directly to the SRG (typically from smaller organisations).
- Approval against the requirements in CAP 670

For both audit and approval, the regulator requires assurance that safety goals will be met. To assist this process the SRG have been developing a goal-based approach to regulation covering software aspects of ATS under the CAP 670 regulations.

3 Goal-based Approach Developed in SW01

At the time of writing, SW01 is in draft form and is being circulated for comment. The first version was produced in conjunction with York Software Ltd (YSL) and subsequent versions were produced with the assistance of Adelard and CSE International Ltd.

These regulatory requirements are intended to be used in conjunction with a higher-level system safety methodology (e.g. as in IEC 61508 [IEC 1998] or ARP 4754 [SAE 1996]). This would cover aspects such as the identification of safety-related functions, system hazards, the system architecture, the allocation of functions to subsystems and the required assurance level.

SW01 sets goals for the assurance needed to show that the software requirements identified from the system safety level analysis have been implemented. In practice there is some iteration between these levels, for example software-level hazards have to be included in the system hazard analysis and this could lead to revised software requirements.

SW01 draws on a number of approaches to implementing a goal-based strategy. The first published implementation of this type of approach was the SafeIT standards framework [Bloomfield 1990] and, more recently, work on safety cases [Adelard 1998] sets out a claim, argument, evidence based structure. Furthermore the UK Ministry of Defence's (MoD) policy paper on Commercial Of the Shelf (COTS) software [MOD 1999] defines a similar policy but as yet there is no detailed guidance on its implementation. The UK Health and Safety Executive (HSE) has sponsored the development of a framework for COTS that also takes a safety argument based approach. Drafts have already been made available for industry comment and the final version should be published early in 2001.

3.1 Safety Goals

SW01 defines a prime software safety objective to be met for ATS systems that contains software and three sub-objectives. These are shown in table 1 below:

Safety Objectives
Prime software safety objective
To ensure that the risks associated with deploying any software used in a safety related ATS system have been reduced to a tolerable level and are As Low As Reasonably Practicable (ALARP).
Sub objectives
To ensure that arguments and evidence are available which show that the Software Safety Requirements correctly state what is necessary and sufficient to achieve adequate safety, in the system context.
To ensure that arguments and evidence are available, in which there is sufficient confidence, which show that the software satisfies its safety requirements.
To ensure that arguments and evidence are available which show that all Safety Requirements can be traced to the design, are realised in the implementation of the software, and that other functions implemented in the software do not adversely affect safety.

Table 1 SW01 Safety Objectives

3.2 The Regulator's Perspective

From the perspective of the regulator, the objective is subtly different, namely:

> For arguments and *assurance evidence* to be available which show that the risks associated with deploying any software used in a safety related ATS system are tolerable and ALARP (As Low As Reasonably Practicable).

In other words it is not enough to produce a safe system, explicit evidence is required to convince the regulator that the risks are tolerable and ALARP. SW01 breaks down this top-level assurance objective into sub-objectives (Table 2) that should achieve the goals; the rationale for the sub-goals is discussed below.

Assurance Objectives	
CONFIGURATION CONSISTENCY	To ensure that the arguments and evidence, for the safety of the software in the system context, are at all times derived from: a known executable version of the software, a known range of adaptation data, and a known set of software products and descriptions that have been used in the production of that version.
REQUIREMENTS VALIDITY	To ensure that arguments and evidence are available which show that the Software Safety Requirements correctly state what is necessary and sufficient to achieve adequate safety, in the system context. (This includes requirements to control hazards identified during implementation)
REQUIREMENTS TRACEABILITY	To ensure that arguments and evidence are available which show that all Safety requirements can be traced to the design, are realised in the implementation of the software**,** and that other functions implemented in the software do not adversely affect safety.
REQUIREMENTS SATISFACTION	To ensure that arguments and evidence are available, in which there is sufficient confidence, which show that the software satisfies its safety requirements.

Table 2 SW01 Assurance objectives

The following sections describe the background to these objectives and the regulatory requirements that flow from them. Figure 1 provides an overview of the

SW01 structure.

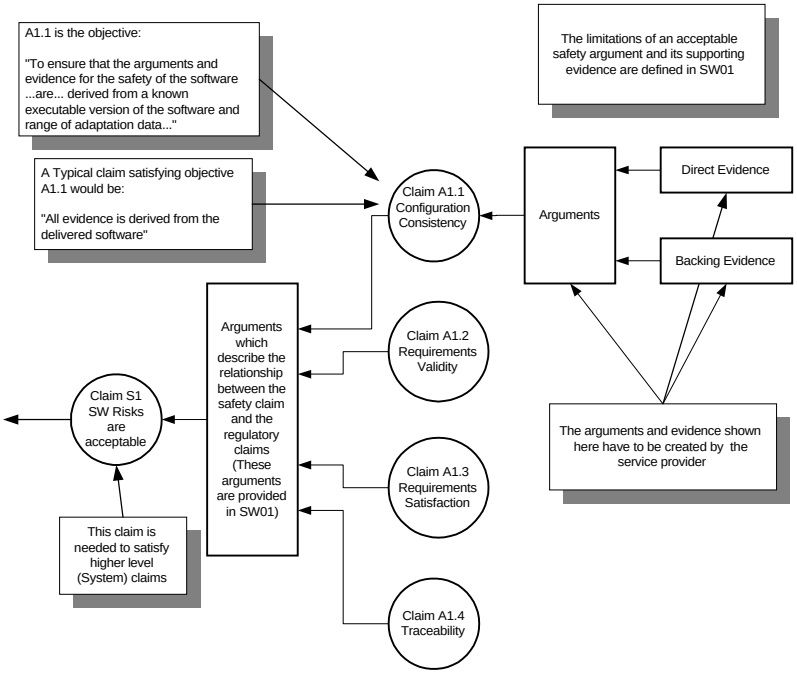

Figure 1 The safety case structure defined by SW01

Configuration consistency

The first regulatory objective is needed because evidence gathered from the development of the product has to be shown to be directly related to the product before it can be used to support an argument about its performance. This objective is called the Configuration Consistency Objective and relates to the wide ranging documentation produced during the software and safety life-cycle. A comprehensive list of development documentation which may be needed to provide evidence is defined in SW01; such artefacts include: source code, object code, system and software requirements, verification data, design data, hazard analyses, data describing the development and test environment, etc.

Requirements validity

Unsafe *behaviour* causes accidents. Consequently, in SW01, the focus on Software Safety Requirements is the behavioural requirements for software. Safety constraints placed upon the software are assumed to be transformed into behavioural requirements during the system safety process.

Each software safety requirement consists of a set of behavioural attributes of interest, these can include: functional properties, timing properties, robustness, reliability, accuracy, resource usage and overload tolerance. Thus a software safety

requirement should define behaviour by specifying the requirement in terms of one or more of these attributes. In fact requirements with complete behaviours are sought, where all relevant attributes are specified in a single requirement. If an attribute is excluded from a software safety requirement an argument is required to justify its exclusion.

There are three main reasons for this stance. Firstly it is probably simpler to analyse the validity and completeness of a requirement if all its attributes are in one place, and one can be reasonably certain that it is independent of other requirements. Secondly, since each requirement describes a complete behaviour and there is a straightforward relationship between behaviour and hazard then the associated AEL can be readily defined. Finally the number of requirements and hence AELs are reduced to a manageable quantity.

During software design (and even during implementation and maintenance) design decisions are made which may have an adverse impact on system safety. For example the choice of an operating system is likely to have a significant impact on the safety claims which, realistically, can be made. Furthermore it is unlikely that the choice of an operating system is made at the system level. In order to preserve the safety properties of the system a hazard analysis should be performed on all design decisions. If a hazard is detected then defences against it propagating and leading to an accident should be defined. Such defences could be procedural, or implemented in hardware or software. In any case the requirement for the defence must be written down. Where the defence is to be implemented in software the requirement is a software safety requirement (a derived requirement) and is considered to be added to the set of Software Safety Requirements which have been delivered to the software development process. Closing this loop provides the assurance that the Software Safety Requirements are complete.

Requirements Satisfaction

SW01 takes the view that software safety comes partly from the correct and complete implementation of those safety requirements which the software implements (called Software Safety Requirements). These Software Safety Requirements can be decomposed using architectural techniques, and the resulting architecture can be assessed (using a knowledge of the system and its operational environment) for its impact on safety, yielding yet more Software Safety Requirements (called Derived Software Safety Requirements). All of these Software Safety Requirements have to be implemented correctly and completely for the system to be considered safe. In SW01 showing that the requirements have been implemented correctly is called Requirements Satisfaction.

Requirements Traceability

The other view of software safety springs from the ease with which functions can be implemented in software and the difficulty in keeping such functions independent of each other. Implementing the Software Safety Requirements correctly and completely is therefore a necessary but insufficient condition for

system safety. In addition, other functions implemented in the software, which are not safety related, must be shown to have no adverse safety impact.

Additionally, during the software development process, functions may be introduced which have repercussions on the safety of the ATS system, even though they may not have been introduced to meet a safety requirement. These will need to be assessed and, if necessary, new or changed Software Safety Requirements will have to be generated. Even if the function requirement has no impact on safety, its implementation may have. Therefore, all the Software Safety Requirements must be traceable to the implementation and all elements of the implementation must be shown to have no adverse safety impact. In SW01 this is called Requirements Traceability.

3.3 Level of assurance

As part of the overall objective that the risks are tolerable and ALARP, the regulator requires assurance evidence that is commensurate with the software risks. In order to indicate to the developer the degree of assurance required by the regulator, we have developed the concept of an *assurance evidence level* (AEL). The AEL is an index running from AEL1 to AEL5—the higher the number, the greater the rigour of the supporting arguments and evidence. AELs are assigned to Software Safety Requirements to identify the depth and strength of software assurance evidence that must be made available for the equipment approval process.

Although the AEL is primarily set by the *safety criticality* of the function (i.e. based on the higher level system safety analysis), in practice the strength and rigour of the evidence required by a regulator will also be dependent on a number of subjective “shaping factors”, i.e.:

- requirements complexity (e.g. complex functionality, stringent timing)
- implementation difficulty (e.g. novelty of solution)
- organisational factors (e.g. complexity of organisational structure, number of subcontractors)

For example, a project under stress, implementing a complex requirement, with a long contractual chain and inexperienced contractors should receive more regulatory oversight than another with a single experience contractor implementing a small system very similar to one that has been successfully implemented in the past. However, we were unable to come up with an acceptable system for describing this in the published document. The AELs have to be established by the ATS service provider and accepted by the regulator.

AELs should not be confused with the SILs defined in IEC 61508.

3.4 Evidence

SW01 identifies two types of assurance evidence that can be used in support of

claims that these objectives have been met:

- *Direct Evidence*: this is evidence that is directly related to the claim being made
- *Backing Evidence*: this is evidence that the direct evidence is both credible and soundly based

Direct evidence is taken from direct measurements of the artefact (note: the artefact could be a design representation of the final product as in static analysis). Other evidence attests to the trustworthiness of the direct evidence or supports an argument of engineering judgement by providing evidence from other sources e.g. equivalent technologies, physical laws. This form of evidence is called "backing" evidence.

Evidence may be gathered from three sources: testing, field service experience and analysis. Here testing is restricted to tests of the final product or a close relation, e.g. host machine tests. Analysis, however, is used in its widest sense and includes inspections and reviews, tests on indirect artefacts such as static analysis and simulations, as well as evidence which supports logical arguments.

Clearly evidence, drawn from a single source, needed to support an argument, must consist of both direct and backing evidence. What is less clear is that a single claim can always be justified by evidence from a single source. As the severity of the potential accidents, to which the software can contribute, increases then it is unlikely that an argument can justifiably be assembled which uses evidence from a single source. Two aspects of evidence underpin this view. Firstly, the evidence may "run out of steam" and not clearly demonstrate that a performance target has been met. This is particularly true of reliability claims, where it is recognised that testing is only convincing below a Mean Time Before Failure (MTBF) of about 10^5 hrs. Secondly, as claims require arguments that are deeper and more rigorous, the potential for "missing" something increases and it is comforting to have independent evidence drawn from more than one source. For this reason a second independent argument (both direct and backing) is expected for the higher AELs—this is termed a Secondary Argument in SW01.

4 Implementation Issues

A goal-based approach, like the one used in SW01, has obvious benefits as it imposes fewer constraints on the implementation, both in terms of processes (e.g. the standards used) and in technical solutions (e.g. the use of off-the-shelf software). The goal-based approach is also useful to the safety assessor, as the questions focus on safety-related outcomes (e.g. "what evidence do you have to show that display updates occur within *x* seconds?).

However, in a safety goal-based approach, it is not sufficient to demonstrate compliance to a generic safety process (such as IEC 61508). Convincing arguments have to be constructed that relate to the behaviour of the specific

product and its safety properties and this can be difficult for contractors to adopt. In other words there is need to shift from documenting how hard people have tried to develop a system, to providing evidence and arguments about the behaviour of that system.

Goal-based safety standards have been used in other sectors (such as parts of the UK Ministry of Defence Standards 00-55 and 00-56). These standards have been used for several years now, but experience indicates that some contractors find it difficult to formulate reasonable arguments or provide convincing evidence. There are a several reasons for this:

- *Limited safety expertise.* Safety expertise is in short supply, and contractors may have to buy-in the skills; such staff may not be aware of the specific sector safety issues or the processes use to produce the product.
- *Lack of integration with design.* Safety is often viewed as a "bolt on" feature, and safety personnel are expected to justify the system after it has been designed. Ideally development of the safety arguments should occur in parallel with the design and influence design choices in favour of solutions where it is easier to make safety arguments.
- *Lack of supporting evidence*. This could be due to design choices (e.g. use of off-the-shelf components with little evidence of integrity) or development deficiencies (e.g. inadequate tests to demonstrate the safety goals or inadequate documentation of fault histories where important data on say operating time has not been collected).

These are not insurmountable , but will require a significant "culture change" to overcome and organisations and people can be slow to adapt to new methods. The key factor is Management commitment to change, but the process can be eased by supporting measures such as:

- *Exchange of experience.* If acceptable safety arguments or specific safety cases can be made available (e.g. via industry safety clubs or safety bodies), this experience could help to train newcomers.
- *Training materials and guidance.* These could include worked examples and templates to show how the safety arguments are identified and supported by evidence. Although in practice it might be hard to find safety cases that are not commercially confidential.
- *Support tools.* The process of constructing and reviewing safety arguments can be assisted by safety argument tools like [ASCE] and [SAM]. For example the ASCE tool allows the structure of the argument to be constructed graphically and can also generate text based reports and display associated safety documents (e.g. in Word or Excel). An example argument exported from ASCE is shown in Figure 2.

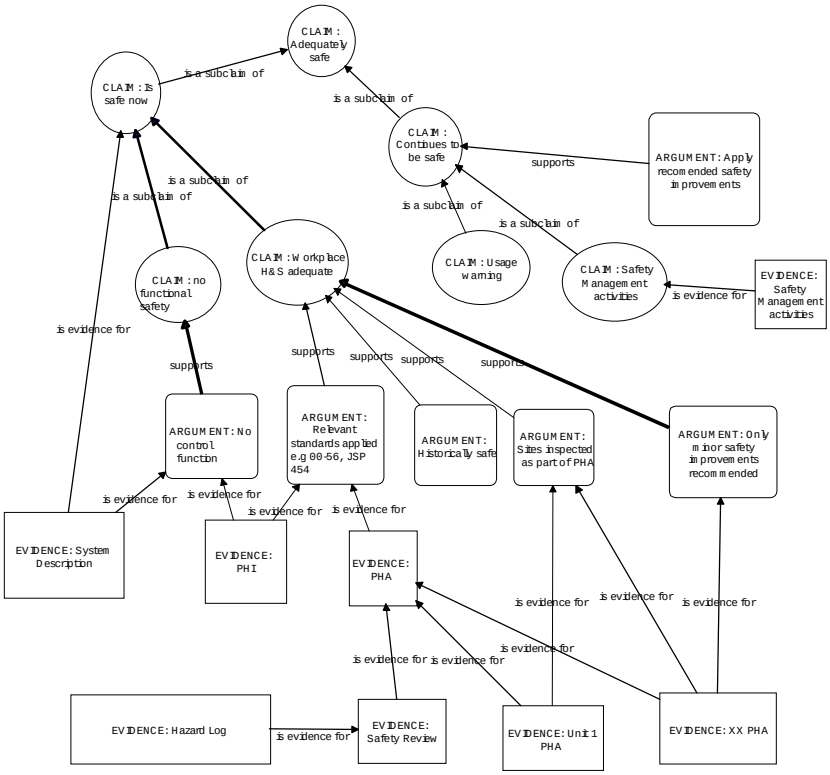

Figure 2. Example argument exported from ASCE

- *Re-usable safety justifications.* The justification effort could be reduced if justifications for software components are re-used over several projects (e.g. an operating system or a database). Similarly if the same system is deployed on different sites similar safety arguments can be deployed, although changes will be needed to take account of the safety requirements of specific sites.

At the other end of the scale, there are experienced organisations with well-established safety management systems (SMS) that have been approved by safety regulators. The safety arguments are developed and maintained internally, but are subject to sample audits by the regulator. In this case there is no lack of expertise, but there is no guarantee the safety justifications will have a goal-based form unless the organisation has explicitly adopted this approach. Nevertheless regulations like SW01 can be used to derive auditing questionnaires to check that the relevant goals are addressed and, in the longer term, the requirements of SW01 could influence the structure of new safety cases as cases compliant with the regulation will be easier to licence. This process could be aided by the supporting measures discussed earlier.

Another aspect that needs to be addressed is the role of standards such as IEC 61508 or RTCA DO178B [RTCA 1992]. Within any given project, different

standards may have been used for different parts (including off-the-shelf components); how does this fit in with a goal-based approach? Demonstration of compliance to a standard is a useful source of evidence for a goal-based approach. In particular, compliance can support claims for configuration consistency, traceability and functional correctness to a given assurance level. However this process evidence is likely to need supplementing with evidence about the product (e.g. via testing, analysis or field experience) that demonstrates the achievement of requirements.

There are also system level aspects to be considered in applying this approach. A goal-based approach can be applied at any level including the top-level system. Based on the criticality of the system and the safety function, safety requirements and assurance levels need to be set for the subsystems. Setting the assurance level is a considerable challenge and needs to take account of factors such as:

- the consequences of system failure
- the degree of mitigation for component failures
- project risk factors such as novel technology and complex requirements

While it is desirable to have a well-defined decision procedure so that the assigned assurance levels are appropriate, excessive assurance would affect project costs and completion times, while insufficient assurance could affect safety. As mentioned above it was not possible in the development of SW01 to find a procedure that was appropriate for a regulation although we have prototyped a number of approaches (scoring methods, decision graphs) that might help.

5 Summary and conclusions

There are clear benefits in adopting a goal-based approach to safety regulation as it gives greater freedom in developing technical solutions and accommodating different standards. A specific example has been described which has been developed for air traffic management software. It explains the reasoning underlying the goal-based approach and the sub-goals and assurance levels required by the regulator and includes features, such as:

- the identification of software safety goals
- a defined relationship between these safety goals and regulatory goals
- an index (AEL) which addresses the consequences of failing to meet a claim and the believability of the claim
- bounds on the acceptability of arguments and evidence

which are specifically related to the goal-based approach.

However, it has to be recognised that such regulations represent a significant “paradigm shift” i.e. a transition from:

- a *process compliance* approach to a *product orientated*, safety property approach
- a *tick-box* mentality to *argument-based* mind-set

Past experience indicates there can be considerable problems in achieving such a culture change within an organisation due to limited safety expertise, poor integration with the design process, and lack of convincing supporting evidence. With these constraints experience indicates that it might take several years to adopt fully. It is therefore important that standards are supported by accompanying measures to facilitate their introduction, such as:

- exchange of experience through special interest groups
- training material and guidance documents
- supporting tools
- re-usable and "model" safety argument examples

These strategies should ease the transition form a prescriptive to a goal-based standards approach.

6 References

[Adelard 1998] ASCAD—Adelard Safety Case Development Manual, ISBN 0 9533771 0 5, Adelard 1998

[ASCE] available from http://www.adelard.co.uk/software/asce/

[Bloomfield 1990] R E Bloomfield, J Brazendale, Standards framework, UK Department of Trade and Industry, June 1990

[CAA 1998] CAP 670 Air Traffic Services Safety Requirements. CAA 1998

[Cullen 1990] Lord Cullen, The public inquiry into the piper alpha disaster. HMSO Cm 1310, 1990

[IEC 1998] Functional safety of electrical/electronic/programmable electronic safety-related systems, IEC 61508, Parts 1 to 7 (1998 onwards – some parts still in draft)

[MoD 1999] Safety Assurance for Non Developmental Safety Critical Software, ES Pol SCS - 23 November 1999, Defence Procurement Agency 1999

[Robens 1972] Lord Robens. Safety and Health at Work. Report of the Committee 1970 - 72. HMSO Cmnd 5034, 1972

[RTCA 1992] RTCA DO178 B - Software Considerations in Airborne Systems and Equipment Certification. RTCA 1992

[SAE 1996] SAE ARP 4754 - Certification Considerations for Highly-Integrated or Complex Aircraft Systems. Society of Automotive Engineers 1996

[SAM] see http://www.cse-euro.demon.co.uk/yse/products/sam/